Integrating Different Functional Modeling Perspectives

Boris Eisenbart, Ahmed Qureshi, Kilian Gericke and Luciënne Blessing

Abstract The paper proposes a modular functional modeling framework, which aims at integrating the different functional modeling perspectives, relevant to different disciplines. The results of two extensive literature studies on diverse functional modeling approaches proposed in a variety of disciplines are consolidated. These studies identified specific needs for an integrated functional modeling approach to support interdisciplinary conceptual design. The presented framework aims at fulfilling these needs. It consists of a variety of associated views, represented through different matrices. This matrix-based representation facilitates the analysis of different functional modeling perspectives and their interdependencies. Finally, the implications of the presented approach are discussed.

Keywords Functional modeling · Functional modeling perspectives · Modeling framework · Interdisciplinary design

B. Eisenbart (✉) · A. Qureshi · K. Gericke · L. Blessing
Engineering Design and Methodology Group, University of Luxembourg, Luxembourg
e-mail: boris.eisenbart@uni.lu

A. Qureshi
e-mail: ahmed.qureshi@uni.lu

K. Gericke
e-mail: kilian.gericke@uni.lu

L. Blessing
e-mail: lucienne.blessing@uni.lu

A. Chakrabarti and R. V. Prakash (eds.), *ICoRD'13*, Lecture Notes in Mechanical Engineering, DOI: 10.1007/978-81-322-1050-4_7,

1 Introduction

Functional modeling is proposed in systematic design approaches across disciplines. It is intended to support early concept development for a technical system, i.e. the transition from a design problem to an early solution concept. Functional modeling results in a first abstract representation of the technical system under development. The term "technical system" encompasses both technical products as well as product/service-systems (PSS) in this paper.

Across and within different disciplines a large variety of function models is proposed and a common approach to functional modeling can hardly be found [1–4]. As a consequence, diverse ways of representing functions are competing when designers from different disciplines collaborate, potentially hindering the exchange expertise [1, 5]. Approaches to bridge the existing diversity, so far, have not been successful [2, 6].

This paper consolidates the results of two extensive literature studies on the diverse functional modeling approaches, proposed in a variety of disciplines [3, 4]. The considered studies identified specific needs for an integrated functional modeling approach linking the different functional modeling perspectives, which are prominently addressed in the proposed function models from different disciplines. This paper presents the concept of an integrated functional modeling approach, which aims at satisfying the identified needs.

2 Towards Integrated Functional Modeling

Eisenbart et al. [3] analyzed function models proposed in mechanical engineering design, electrical engineering design, software development, mechatronic system development, service development and PSS design. The particular content addressed by individual function models is linked to different functional modeling perspectives. Seven central perspectives have been identified, which are described in Table 1.

None of the reviewed function models from the different disciplines addresses all identified functional modeling perspectives [3]. In each considered discipline a different set of functional modeling perspectives is prominently addressed. However, Eisenbart et al. [4] identified the *transformation process perspective* to be prominently addressed in functional modeling approaches proposed across all reviewed disciplines. It may, hence, serve as a common basis in an integrated functional modeling approach. Based on the two literature studies by Eisenbart et al.[1] specific needs for such a modeling approach can be formulated. Accordingly, an adequate integrated functional modeling approach should:

[1] Eisenbart et al. considered 70 function models (54 original models plus variants proposed by different authors) and 41 systematic design approaches. The respective references may be taken from [3, 4].

Table 1 Functional modeling perspectives, after [4]

States	Representation of the states a system can be in, or of the states of operands before (input) and after (output) a transformation process. Operands are typically specifications of energy, material, and information
Effects	Representation of the required physiochemical effects, which have to be provided to enable, respectively support, the transformation process(es) changing one state into another state
Trans-formation processes	Representation of the processes executed by stakeholders or technical systems, which (from the designers' perspective) are part of the technical system under development in order to change the state of the system or of operands. *Technical processes* are transformation processes related to technical systems, while *human processes* are related to stakeholders (thus, including service activities)
Interaction processes	Representation of interaction processes of stakeholders or of other technical systems, which (from the designers' perspective) are *not* part of a system, with stakeholders or technical systems, which *are* part of the system under consideration
Use case	Representation of different cases of applying the technical system. This is typically associated to the interaction of stakeholders or another technical system with the technical system under development, which triggers, respectively requires subsequent processes to take place
Technical system allocation	Representation of the role of a technical system, which is supposed to perform or enable a (sub-) set of required *effects* or *processes*, either as part of the technical system under consideration or by interacting with it
Stakeholder allocation	Representation of the roles of different stakeholders, which may be users benefitting from a system or operators contributing to the system, e.g. through executing required processes or providing resources, etc

- …link the identified functional modeling perspectives, in order to relate between information, which is relevant to the designers from the different disciplines.
- …enable flexibly switching between considered functional modeling perspectives, in order to facilitate adaption of the modeling approach to different design approaches.
- …provide a condense and clearly structured representation, in order to ease comprehension of the modeled functions among collaborating designers. Often multiple complementary models are proposed in a functional modeling approach. Comprehensively capturing information distributed across different models can be a difficult cognitive task. However, *one* model covering a large number of functional modeling perspectives may quickly become confusingly packed with information.
- …facilitate linking functions in different ways, in order to be adaptable to discipline-specific representations. Depending on the particular discipline, functions may essentially be linked related to *time* (particularly prominent in software and service development), *input/output relations* (particularly prominent in mechanical engineering design) or *hierarchy*.

- …address impacts from, respectively on the environment, in order to facilitate finding viable solution concepts [7]. Only few authors explicitly consider the environment within functional modeling (e.g. [8, 9]). However, impacts from the environment on a technical system may impair function fulfillment. In turn, impacts from a technical system on the environment may be critical to e.g. safety requirements or environmental legislation.

Beyond these needs, additional options are discussed by Eisenbart et al. [3], which may considerably support the reasoning about functions within system conceptualization; such as considering function-sharing, the inclusion of quantities, as well as a stronger link between the functional model of a system and its structure.

3 Integrated Functional Modeling Framework

In order to meet the needs discussed in the previous section, this paper proposes the integrated functional modeling framework (IFM framework), which aims at supporting integrated modeling of the identified functional modeling perspectives.

3.1 Development of the IFM Framework

In the development of the functional modeling framework, different alternatives have been generated. Firstly, an attempt was made to adapt existing functional modeling approaches to satisfy the specific needs discussed above. For this, several approaches have been selected, which already cover a large variety of the different functional modeling perspectives. Each generated alternative has been applied for re-modeling examples of existing function models from the literature as well as an example from industry. The generated models and approaches have been comparatively evaluated.

From the authors' point view, merely expanding existing approaches has not resulted in suitable integrated modeling approaches: The respective models frequently seemed overburdened with the represented information and thus quickly became very difficult to comprehend. Often, the dependencies between the different functional modeling perspectives in relation to the central *transformation process perspective* could not adequately be addressed. Also, the link between individual functional modeling perspectives often became fuzzy, with the result that individual perspectives could hardly be reasoned upon disconnected from others.

Existing function models typically use blocks or circles for depicting transformation processes, states, effects, etc. These elements are typically arranged *circular* or in *vertical/horizontal* flows. The functional modeling framework presented in the following, instead, uses a modular, *matrix-based* representation, which allows modeling and retrieving information more clearly. The developed approach is related to the concept of multi-domain matrices (MDM) proposed by Lindemann and Maurer [10]. MDM map different design information, in order to facilitate analysis and representation of interdependencies. Similarly, the IFM framework aims at clearly representing information associated to the individual modeling perspectives and their dependencies.

3.2 Outline of the IFM Framework

The IFM framework consists of associated modular matrices representing different views onto the functions of a system under development. The central view (*process flow view*) addresses the *transformation process perspective*, which is prominent in functional modeling approaches across disciplines. Associated views use matrices to represent information about the different entities and their interdependencies in the modeling framework. The entities are directly linked to the specific functional modeling perspectives discussed above (see Table 2).

The framework of modular, adjacent views provides a clearly structured representation and allows taking different views on the functions of a technical system. This modular structure allows addition or omission of views related to the specific needs of the involved designers. The following sub-section describes the entities and their relations, which form the basis of the IFM framework. Section 3.2.2 describes the associated views, which form the representation of the IFM framework.

3.2.1 Entities and their Relations

The class diagram in Fig. 1 represents the relations between individual entities in the developed modeling framework. A technical system under development may support one or more use cases. Each use case may be decomposed into sub-use cases. Use cases may have dependencies among each other that may be bound by

Table 2 Entities in the IFM framework and addressed functional modeling perspectives

Entity	Addressed functional modeling perspective
Use case	Use case perspective
State	States (operands and system)
Process	Transformation process and interaction process perspectives
Effect	Effect perspective
Actor	Stakeholder and technical system allocation perspective; system state perspective
Operand	Operand state perspective

specific constraints (mutually exclusive, mutually inclusive etc.,). For all other situations, in Fig. 1, the dependencies shown will be used to depict the similar constraints.

A use case may have one or more transformation process associated to it. There may be dependencies between individual transformation processes, which may or may not be also composed of sub-processes. A transformation process results in the transformation of one or more operand and/or actor from a given state into another. Such state transformations are enabled, respectively supported by effects, which are provided by actors. Actors, by providing the necessary effects, act as operators in transformation processes. Actor is a super class which contains the sub classes of stakeholder, technical (sub-) system, and environment. The actor sub-class of stakeholder comprises (groups of) animate beings affected by or affecting the technical system under consideration (including any related services). The actor sub-class of technical (sub-) system encompasses technical systems which are sub-systems to the technical system under development. It can also be composed of more technical (sub-) systems. Actors also may have dependencies among each other. Environment includes all active and passive parts of nature in general surrounding the system under development.

3.2.2 Associated Views

The different views are strongly linked to each other through the adjacent placement and the respectively shared header rows and header columns in the specific matrices forming the individual views (see Figs. 2, 3, 4, 5). The aim behind this specific set-up is to interlink all the different functional modeling perspectives (i.e. the corresponding views), prominent in the different disciplines, via the *transformation process perspective* (i.e. the central *process flow view*), which is

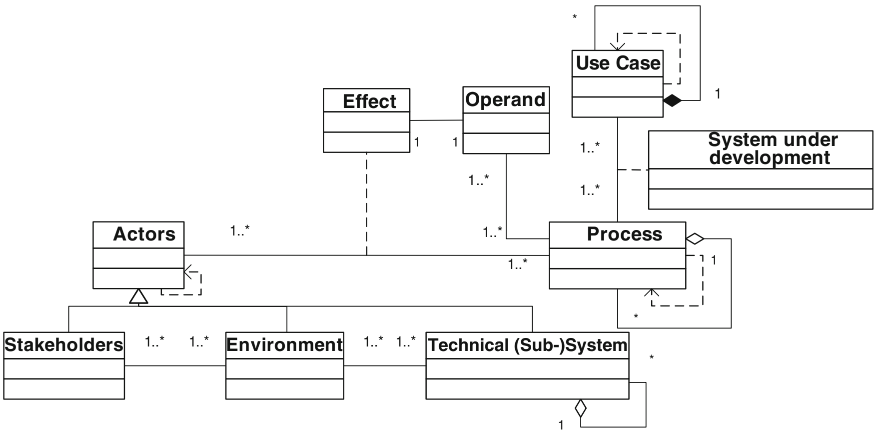

Fig. 1 Class diagram of the developed functional modeling framework

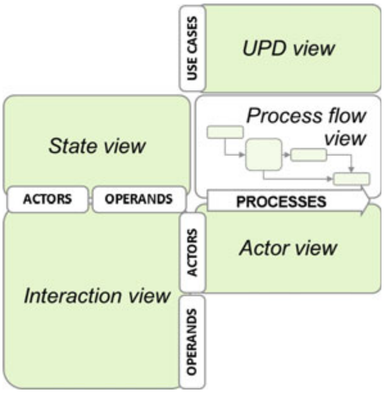

Fig. 2 Adjacent views in the IFM framework

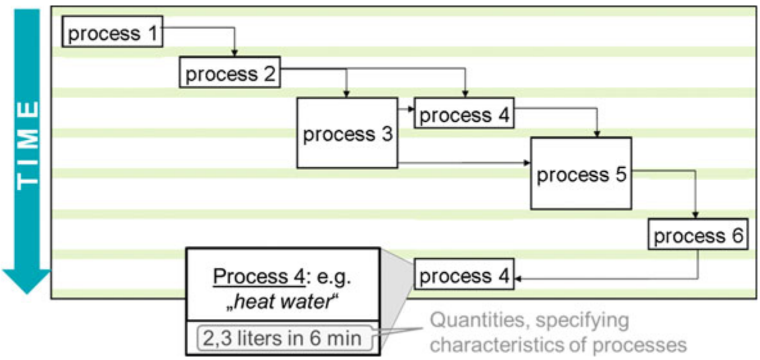

Fig. 3 Process flow view

prominent across disciplines. The individual views and how they link to each other are described in the following.

The *Process flow view* visualizes the flow of processes related to a specific use case. In the view individual processes are represented as chronologically numbered blocks. In the vertical direction, the process flow related to time is visualized. The flow qualitatively illustrates whether individual processes are expected to be carried out sequentially, in parallel or to be overlapping with other processes. The process blocks are furthermore spread horizontally from left to right, so as to enable a direct link to the *actor view* matrix, which is described further down. As an option, quantities related to individual processes can be included to specify processes further, as illustrated in Fig. 3.

The *Effect view* represents the effects, which enable individual transformation processes and are provided by actors. For each process block in the *process flow view*, a separate *effect view* may be created. Similar to the *process flow view*,

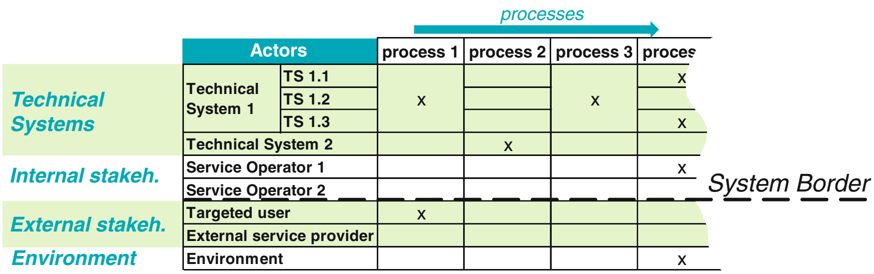

Fig. 4 Actor view matrix

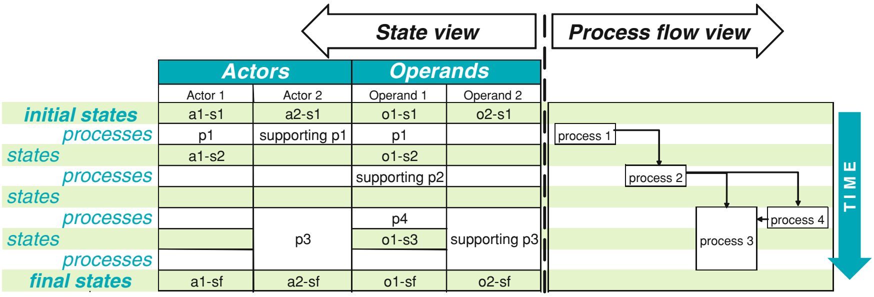

Fig. 5 System states view associated to process flow view

effects can be modeled related to time or flow of operands. Hierarchical trees or alternative models (similar to [7, 8]) may also be applied.

The *Use case/process dependency* (*UPD*) *view* indicates the involvement of individual processes within different use cases. The individual use cases are listed in the header column. The matrix is directly linked to the *process flow view*. The individual—strictly horizontally ordered—process blocks build up the header row for the *UPD view* (see Fig. 2). Dependencies between use cases and processes could affect their operability. For instance, the processes of "heating water" in one use case and the process of "cooling water" in another use case should not be executed in parallel for the same water sample; hence, neither should the respective use cases.

The *Actor view* indicates the involvement of specific actors in the realization of transformation processes. Transformation processes are spread in the header row, associated to the process flow view (see Fig. 2). Within the matrix, involvement may initially be indicated with an "x". As more information becomes available in the design process, the particular role of actors (e.g. as either "affecting" or "being affected" by a process) can be more concretely specified.

The *actor view* allows differentiating actors according to whether they—from the designers' point of view—are part of the system under development (e.g. service operators as part of a PSS) or not (e.g. the targeted users or external service providers). This differentiation is particularly important in PSS design [11]. Through this differentiated allocation, individual processes are separated between *transformation processes* (enabled by actors which are part of the system) and *interaction processes* (enabled by actors which are not part of the system).

The *States view* represents the specific states of operands and agents as well as the state changes caused by individual processes. The *system states view* consists of the *actor state matrix* and the *operand state matrix*, and is a modular addition to the *process flow view* (see Figs. 2 and 5). The adjacent placement of the *system state* and *process flow views*, as shown in Fig. 5, allows the development of the views and the verification of their consistency. Considering the required changes from initial to final states of operands and actors facilitates the development of the *process flow view* and vice-versa. The system state view also allows the indication of operands supporting a transformation process without changing their states.

Figure 5 illustrates, e.g., that a process "heat water" (process 1) is linked to a change of the technical system (actor 1) from switched-off (a1-s1) to switched-on (a1-s2), supported by an operator (actor 2), as well as a change of the state of water (operand 1) from liquid (state o1-s1) to steam (new state o1-s2). The steam may then be used to drive a turbine (process 2), which may give rise to changes to other operands and/or actors. During this process, the state of the water (i.e. steam) is not changing, but supporting process 2.

The *interaction view* depicts the specific interactions between actors and operands, as well as among each other, in the realization of processes. The view uses operands and actors as both heading column and heading row, as illustrated in Figs. 2 and 6. The specification of the interaction between actors or operands includes the number of the respective process (to provide clarity, as numerous interactions may occur related to different processes) and a short statement specifying the interaction. Analysis of interferences between actors and/or operands may highlight problems with function fulfillment. Also, information about how actors and operands may impact on each other facilitates the design of the interfaces between them accordingly in later design phases.

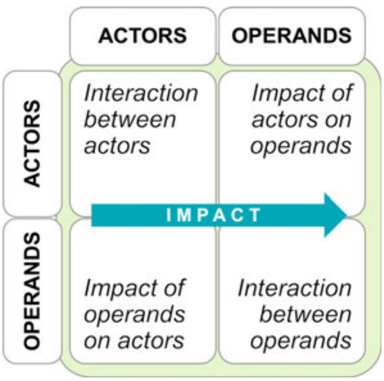

Fig. 6 Concept of interaction view matrix

To give an example, a stakeholder (actor) may have an impact on a technical system (actor) by pushing a button. Similarly, hot water (operand) may impact on a technical system (actor) through transmitting heat or vise-versa. Finally, operands may also impact on each other, as for instance, cold water (operand) may be used to cool hydraulic fluids (operand) and vise-versa. In case, the specifics of an interaction cannot be specified at an early point in the modeling process, the respective cell may initially be marked with an 'x'. Optionally, information about how the interaction is embodied may be included; such as 'mechanical contact' between the stakeholder's finger and the button being pushed.

3.3 Application

The presented framework may be applied in different ways, i.e. depending on the specific approach taken by designers, alternative entry points and sequences of steps may be applied. One potential sequence of modeling activities for an original design project is described in the following. Starting point may be a comprehensive requirements specification (or similar).

- Step 1—*Use Case definition* includes the consolidation of the different use cases (and their sub-use cases, if applicable) the system under development is expected to support in the different phases of its life-cycle. The use cases are represented in the respective column in the *UPD view*.
- Step 2—*Process flow modeling* involves modeling separate flows of required transformation and interaction processes related to each (sub-) use case. A multitude of alternative process flows may fulfill a use case. As described above, modeling and selecting an alternative process flow may be facilitated through considering the required state changes of (supporting) operands in the *operand states matrix* (as part of the *state view*). While modeling the process flows, the involvement of processes in multiple use cases (represented in the *UPD view*) needs to be considered.
- Step 3—*Effect modeling* involves modeling the required effects related to the specific process flows. Considering the basic required effects enabling transformation processes may considerably support the allocation of actors in the following step.
- Step 4—*Actor allocation* includes allocation of the actors, which are involved in the individual processes, either as affecting or being affected through the delivered effects. Actor allocation may be supported through applying the function-means pattern, morphological charts or similar approaches. Carefully considering re-use of allocated actors in different processes and use cases may facilitate function sharing.
- Step 5—*State modeling* includes modeling the state changes of allocated actors in the actor state matrix (as part of the *state view*) related to the chosen process flows.

- Step 6—*Interaction specification* involves analyzing and detailing the specific interactions (i.e. the bilateral impacts) among actors, among operands, and between actors and operands in the realization of processes.

There can be iterations within and between individual steps. For instance, depending on the specific choice of realizing actors, the chosen process flows may have to change, requiring iterations between steps 2–4. *Actor allocation* essentially marks the transition from the problem to the solution. However, the final set of *process flow view* and *actor view* merely represents one potential concept out of large number of variants.

Modeling starts on a high level of abstraction defining the use cases, associated processes etc. On the next level of detail, individual process blocks may then be regarded as use cases comprised of sub-processes. These are enabled by technical (sub-) systems (which may again be comprised of general function carriers or "organs" [8], which are gradually concretized) including any related service operator etc. Thus, the framework allows modeling the functions and actors of a system under development from very abstract to very detailed and concrete.

4 Discussion and Conclusion

Functional modeling is proposed across disciplines to support early concept development. The different functional modeling perspectives prominently addressed in the different disciplines need to be integrated in order to support interdisciplinary functional modeling. In this paper, an integrated modeling framework has been proposed, which aims at linking the different functional modeling perspectives. The proposed framework uses interlinked modular matrices, representing different views on the functions of a system under development. The different views represent individual functional modeling perspectives and/or dependencies between them. It is expected to provide designers from different disciplines with a valuable approach to functional modeling, as it.

- uses an established matrix-based approach for analyzing and representing interdependencies between functional modeling perspectives, similar to MDM;
- considers all identified functional modeling perspectives and their interdependencies;
- is expected to ease communication across disciplines, as the different views are linked via a central view, which is commonly prominent across disciplines;
- is modular, which enables addition or omission of views and related modeling activities depending on whether these are needed in a specific design context;
- allows using different views separately; the designers may flexibly switch between considered views, which allows focusing on specific functional modeling perspectives;

- the strong links between the modular matrices representing the different views provide a clearly structured representation supporting comprehension of complex systems;
- allows embedding existing (discipline-specific) function models[2];
- is open for existing functional taxonomies to be embedded;
- can address the functions of a technical system on different level of detail/abstraction;
- is expected to be easily transferrable into a software tool;
- finally, integrated the consideration of the environment.

In summary, the proposed functional modeling framework aims at fulfilling the formulated needs and—through its specific structure—is expected to support the exchange of discipline-specific expertise during system conceptualization. The explicit inclusion of the specific interactions between individual actors is further expected to provide links to models used in subsequent design phases (e.g. system structure, interface matrix, etc.,). Apart from the presented views, the framework may be further expanded. For instance, additional views may address the dependencies among different states (for both operands and actors), different use cases, different processes (across use cases) etc.

Future research will address the practical application of the developed framework by designers in industry. That will include workshops, wherein practical designers from different disciplines apply the developed framework in conceptual design of mechatronic systems and PSS. It will be of particular interest which specific functional modeling perspectives are most relevant to designers from different disciplines and how designers reason between the different proposed views in different design contexts. The gained insights and feedback from the designers will be used to develop the framework further, in order to improve its applicability in different design contexts.

Acknowledgments The authors would like to thank the Fonds Nationale de la Recherche Luxembourg for funding this research as well as Prof. Mogens Myrup Andreasen (DTU Copenhagen), Prof. Pierre Kelsen (Luxembourg University), and Hubert Moser (LuxSpace) for valuable feedback on earlier versions of the presented functional modeling framework.

References

1. Erden M, Komoto H, van Beek TJ, D'Amelio V, Echavarria E, Tomiyama T (2008) A review of function modeling—approaches and applications. Artif Intell Eng Des Anal Manuf 22:147–169

[2] This has only been briefly discussed related to the description of the *effect view*. However, numerous other cases wherein alternative existing models may be applied (either complementing or in exchange for matrices associated to a specific view) can be thought of.

2. Vermaas P (2011) Accepting ambiguity of engineering functional descriptions. Proceedings of 18th international conference on engineering design, ICED, pp 98–107
3. Eisenbart B, Blessing LTM, Gericke K (2012) Functional modeling perspectives across disciplines—a literature review. Proceedings of 12th international design conference, design
4. Eisenbart B, Gericke K, Blessing L (2012) A shared basis for functional modeling. Proceedings of 9th Nord design conference
5. Müller P, Schmidt-Kretschmer M, Blessing L (2007) Function allocation in product-service-systems—are there analogies between PSS and mechatronics? AEDS Workshop
6. Carrara M, Garbacz P, Vermaas P (2011) If engineering function is a family resemblance concept. Appl Ontol 6:141–163
7. Pahl G, Beitz WFJ, Grote K-H (2008) Engineering design—a systematic approach. Springer, Berlin
8. Hubka V (1984) Theorie technischer systeme—grundlagen einer wissenschaftlichen konstruktionslehre. Springer
9. Watanabe K, Mikoshiba S, Tateyama T, Shimomura Y, Kimita K (2011) Service design methodology for cooperative services. Proceedings of the ASME IDETC/CIE
10. Lindemann U, Maurer M (2007) Facing multi-domain complexity in product development. In: Krause FL (ed) The future of product development, Springer, Berlin
11. Tan AR (2010) Service-oriented product development strategies